

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

Claim 1 (currently amended): A system for managing persistent objects for an application accessing data, wherein said persistent objects are stored in at least one data source, comprising:

a persistent object framework to provide data from and perform functions on said persistent objects in accordance with said application, wherein the persistent object framework is between the application and the data source and providing an interface for the application to query the data source; and

a cached set of persistent objects within said persistent object framework and managed by the persistent object framework, the persistent objects being identified for use by said application and corresponding to said stored persistent objects.

Claim 2 (original): The system of claim 1, wherein said application is a Java servlet.

Claim 3 (currently amended): The system of claim 1, wherein said functions include a function to create a persistent object, the creation function comprising caching the newly-created persistent object in the cached set of persistent objects and then, inserting the newly-created persistent object into the data source after a save transaction has been committed or a flush method has been invoked.

Claim 4 (original): The system of claim 1, wherein said functions include a function to cache a persistent object.

Claim 5 (currently amended): The system of claim 1, wherein said functions include a function to update a persistent object, wherein the update function comprises deferring comprises writing to a persistent object in the data source that is deferred until a transaction is committed, a flush method is invoked, or a query process is started.

Claim 6 (original): The system of claim 1, wherein said persistent object framework includes a set of data models corresponding to said stored persistent objects.

Claim 7 (currently amended): The system of claim 1, further comprising an additional data source and wherein said persistent object framework includes an object space to map said persistent objects to ~~locations within said at least one data source~~ the at least one data source or the additional data source based on a type definition for each of the persistent objects.

Claim 8 (original): An application system supported by a Java programming environment, comprising:

AS a relational database storing a first set of persistent objects correlating to an application;

a LDAP repository storing a second set of persistent objects correlating to said application; and

a persistent object framework to provide data to said application from said first and second sets of persistent objects, wherein said persistent object framework caches a subset of said first and second persistent objects from said relational database and said LDAP repository.

Claim 9 (original): The application system of claim 8, further comprising JDBC services coupling said persistent object framework with said relational database.

Claim 10 (original): The application system of claim 8, further comprising business objects identified by said application and corresponding to said stored sets of persistent objects.

Claim 11 (original): The application system of claim 8, wherein said persistent object framework includes a set of data models to create said persistent objects.

Claim 12 (currently amended): A method for managing persistent objects correlating to an application, comprising:

mapping a persistent object stored within a data source with a persistent object framework coupled to said application;

identifying said persistent object stored in said data source as applicable to said application; [[and]]

caching said persistent object within a cache managed by said persistent object framework[.]; and

with the persistent object framework, creating a new persistent object according to a data model stored by the persistent object framework, the creating comprising caching the new persistent object in the persistent object cache and inserting the new persistent object in the data source after a save transaction has been committed or a flush method has been invoked.

Claim 13 (canceled)

Claim 14 (currently amended): The method of claim [[13]] 12, wherein said creating includes determining initial values for attributes within said data model.

Claim 15 (currently amended): The method of claim [[13]] 12, wherein said creating includes updating a revision indicator within said data model.

Claim 16 (currently amended): The method of claim ~~[[13]]~~ 12, wherein said creating includes determining a persistent object identity for said newly created persistent object.

Claim 17 (currently amended): The method claim ~~[[13]]~~ 12, wherein said creating includes defining subtypes for said newly created persistent object.

Claim 18 (original): The method of claim 12, further comprising saving said persistent object by said persistent object framework.

Claim 19 (original): The method of claim 12, further comprising accessing said persistent object by said persistent object framework.

Claim 20 (original): The method of claim 12, further comprising updating said persistent object by said persistent object framework.

Claim 21 (original): The method of claim 12, further comprising deleting said persistent object by said persistent object framework.

A8
Claim 22 (currently amended): A method for searching persistent objects stored in at least one data source, wherein an application accesses said persistent objects for data, comprising:

receiving a search query for a persistent object at a persistent object framework;

determining a query type for said search query, wherein said query type is selected from the group of query types consisting of a primary key, a handle, a unique key, a query filter, and a relationship between persistent objects;

when said query type is determined not to be a query filter, searching a cache within said persistent object framework for said persistent object according to

said query type; and

searching said data source when said persistent object is not within said cache or when said query type is determined to be a query filter.

Claim 23 (currently amended): The method of claim 22, wherein said ~~second searching step~~ searching of said data source includes enabling a lazy load state.

Claim 24 (original): The method of claim 23, further comprising determining whether said persistent object data is needed by said application.

Claims 25-29 (canceled)

Claim 30 (original): The method of claim 22, further comprising discarding said search query.

Claims 31 and 32 (canceled)

A8
Claim 33 (original): A method for resolving a stale data state between a persistent object and an application accessing said persistent object for data, comprising:
executing a process of said application accessing said persistent object,
wherein said persistent object includes a revision attribute;
identifying said stale data state within said persistent object by a persistent object framework,
retrying said process of said application accessing said persistent object;
and
incrementing said revision attribute.

Claim 34 (original): A method for managing persistent objects within an application system, wherein said persistent objects are stored within a first data source and a second data source and said persistent objects provide data to an application, comprising:

implementing a persistent object framework that caches said persistent objects correlating to said application by:

creating said persistent objects;
caching said persistent objects;
accessing said persistent objects;
updating said persistent objects;
searching said persistent objects;
deferring writes to said first and second data sources; and
controlling persistent storage of said persistent objects; and

retrieving said data from said first and second data sources when requested by said persistent object framework.

A8
Claim 35 (currently amended): A system for managing persistent objects correlating to an application, comprising:

means for mapping a persistent object stored within a data source with a persistent object framework coupled to said application;

means for identifying said persistent object stored in said data source as applicable to said application; [[and]]

means for caching said persistent object within said persistent object framework[[]] ; and

means for creating a new persistent object according to a data model stored by the persistent object framework, the creating comprising caching the new persistent object in the persistent object cache and inserting the new persistent object in the data source after a save transaction has been committed or a flush method has been invoked

Claim 36 (currently amended): A computer program product comprising a computer useable medium having computer readable code embodied therein for managing persistent objects correlating to an application, the computer program product adapted when run on a computer to effect steps, including:

mapping a persistent object stored within a data source with a persistent object framework coupled to said application;

identifying said persistent object stored in said data source as applicable to said application; [[and]]

caching said persistent object within said persistent object framework[[]] ;
and

creating a new persistent object according to a data model stored by the persistent object framework, the creating comprising caching the new persistent object in the persistent object cache and inserting the new persistent object in the data source after a save transaction has been committed or a flush method has been invoked.

A8
Claim 37 (currently amended): A system for searching persistent objects stored in at least one data source, wherein an application accesses said persistent objects for data, comprising:

means for receiving a search query for a persistent object at a persistent object framework;

means for determining a query type for said search query, wherein said query type is selected from the group of query types consisting of a primary key, a handle, a unique key, a query filter, and a relationship between persistent objects;

when said query type is determined not to be a query filter, means for searching a cache within said persistent object framework for said persistent object according to said query type; and

means for searching said data source when said persistent object is not within said cache.

Claim 38 (currently amended): A computer program product comprising a computer useable medium having computer readable code embodied therein for searching persistent objects stored in at least one data source, wherein an application accesses said persistent objects for data, the computer program product adapted when run on a computer to effect steps, including:

receiving a search query for a persistent object at a persistent object framework;

determining a query type for said search query, wherein said query type is selected from the group of query types consisting of a primary key, a handle, a unique key, a query filter, and a relationship between persistent objects;

when said query type is determined not to be a query filter, searching a cache within said persistent object framework for said persistent object according to said query type; and

searching said data source when said persistent object is not within said cache.

Claims 39 and 40 (canceled)

AS
Claim 41 (currently amended): A system for resolving a stale data state between a persistent object and an application accessing said persistent object for data, comprising:

means for executing a process of said application accessing said persistent object wherein said persistent object includes a revision attribute;

means for retrying the process of said application accessing said persistent object; and

means for incrementing said revision attribute.

means for identifying said stale data state within said persistent object by a persistent object framework[[,]] .

Claim 42 (original): A computer program product comprising a computer useable medium having computer readable code embodied therein for resolving a stale data state between a persistent object and an application accessing said persistent object for data, the computer program product adapted when run on a computer to effect steps, including:

executing a process of said application accessing said persistent object wherein said persistent object includes a revision attribute;

identifying said stale data state within said persistent object by a persistent object framework,

retrying a second process of said application accessing said persistent object; and

incrementing said revision attribute.

Claim 43 (original): A system for managing persistent objects within an application system, wherein said persistent objects are stored within a first data source and a second data source and said persistent objects provide data to an application, comprising:

means for implementing a persistent object framework that caches said persistent objects correlating to said application comprising:

means for creating said persistent objects;

means for caching said persistent objects;

means for accessing said persistent objects;

means for updating said persistent objects;

means for searching said persistent objects;

means for deferring writes to said first and second data sources; and

means for controlling persistent storage of said persistent objects; and

means for retrieving said data from said first and second data sources when requested by said persistent object framework.

Claim 44 (original): A computer program product comprising a computer useable medium having computer readable code embodied therein for managing persistent objects within an application system, wherein said persistent objects are stored within a first data source and a second data source and said persistent objects provide data to an application, the computer program product adapted when run on a computer to effect steps, including:

implementing a persistent object framework that caches said persistent objects correlating to said application comprising:

A8
creating said persistent objects;
caching said persistent objects;
accessing said persistent objects;
updating said persistent objects;
searching said persistent objects;
deferring writes to said first and second data sources; and
controlling persistent storage of said persistent objects; and

retrieving said data from said first and second data sources when requested by said persistent object framework.
